
Enlightenment Thinkers Network Analysis - Code Cheat Sheet

Build a Simple Network Analysis Program

Topics in Humanities Fall 2025 Instructor: C. Hurst

Step 1: Import Tools

```
# Load tools: data handling + network building + plotting
import pandas as pd      # "pd" = handle tables (nodes/edges)
import networkx as nx    # "nx" = build networks
import matplotlib.pyplot as plt # "plt" = draw graphs
```

Step 2: Load Humanities Data

```
# Read node/edge tables (save files as CSV in the same folder!)
nodes = pd.read_csv("nodes.csv") # Thinkers: id, name, field (e.g., Philosophy)
edges = pd.read_csv("edges.csv") # Relationships: source, target, relationship
```

Step 3: Build the Network

```
G = nx.Graph() # Create empty network

# Add thinkers (nodes)
for _, row in nodes.iterrows():
    G.add_node(
        row["id"],          # Unique ID (e.g., 1 = Voltaire)
        name=row["name"],   # Thinker's name
        field=row["field"] # Field (e.g., Philosophy)
    )

# Add relationships (edges)
for _, row in edges.iterrows():
    G.add_edge(
        row["source"],      # Who (ID)
        row["target"],      # Connected to (ID)
        relationship=row["relationship"] # How (e.g., "Friend")
    )
```

Step 4: Add a New Thinker

```

# Add Kant (Philosophy) + "Admirer" relationship to Voltaire (ID=1)
G.add_node(9, name="Kant", field="Philosophy")
G.add_edge(9, 1, relationship="Admirer")

```

Step 5: Draw the Network

```

# Position nodes neatly (3D: add dim=3)
pos = nx.spring_layout(G, k=2) # k = node spacing

# Color nodes by field (e.g., Philosophy=red)
field_colors = {"Philosophy": "red", "Politics": "blue", "Literature": "green"}
node_colors = [field_colors[G.nodes[node]["field"]] for node in G.nodes]

# Draw nodes/edges/labels
nx.draw_networkx_nodes(G, pos, node_color=node_colors, node_size=1500)
nx.draw_networkx_edges(G, pos, edge_color="gray")
nx.draw_networkx_labels(G, pos, labels={n: G.nodes[n]["name"] for n in G.nodes})
nx.draw_networkx_edge_labels(G, pos, edge_labels=nx.get_edge_attributes(G, "relatic

plt.axis("off") # Hide messy axes
plt.show()     # Show your network!

```